# Mining Frequent Patterns, Associations, & Correlations

- **Motivation**: Business transaction records
    - Discovery of *interesting correlation relationships* that help *business decision-making processes* (catalog design, cross-marketing, customer shopping behavior analysis, …)

اللي خلانا نعمل فكرة mining frequent هو السوبر ماركت في الحالات العاديه الزبون بيدخل السوبر ماركت وبيشتري الحاجات اللي عايزها وبيدفع تمنها ويمشي وملهوش علاقي باللي بيحصل بعد كده المفروض بقي صاحب السوبر ماركت يشوف ايه اكتر حاجات اتباعت ويحطهم جنب بعض زي الشاي والسكر او العيش واللبن ويبدا يعيد توزيعهم جنب بعد ولو لقي حاجه معينه محدش بيشتريها كتير ممكن يحطها جنبهم ويحدد يوم ايه اللي بيحصل فيه كده زي الاجازات كده ينزل الخصومات دي في يوم الاجازه بحيث يسهل عمليه البيع

Cross marketing احط مثلا الموبايل جنب الاكسيسورات بتاعته عشان ازود فرصه شرائهم سوا

Market basket analysis

مين اشتري ايه واشتري معاها ايه

- How to place SW, HW, and Accessories?

ودي علي حسب الشخص اللي بيبيع والمبيعات عنده يعني لو لقي HW , accessories بيتباعوا كتير ممكن يحط جنبهم SW عشان يزود نسبه بيعها يعني مسائله تقديريه

- Frequent patterns are item sets that appear frequently in a dataset (e.g. transaction records)

مجموعه items اللي بتتكرر مع بعضها كتير

*Frequently associated* ممكن اطلع منها association rules يعني لو اشتريت كمبيوتر يبقي لازم اشتري انتي فيرس(اينعم احنا بنجيبه مسروق من النت بس ده في الدول المتقدمه)

- Support and Confidence are measures of <u>rule interestingness</u>

Support and Confidence بتحددلي مدي ثقتي في rule اللي انا حددتها دي

    - 2% support → 2% of transactions show that computers and AV_SW are bought together بشوف الحاجتين دول اتباعوا سوا
    - 60% confidence → 60% of customers who bought a computer also bought AV_SW ان الزبون اللي اشتري كمبيوتر اشتري معاه انتي فيرس

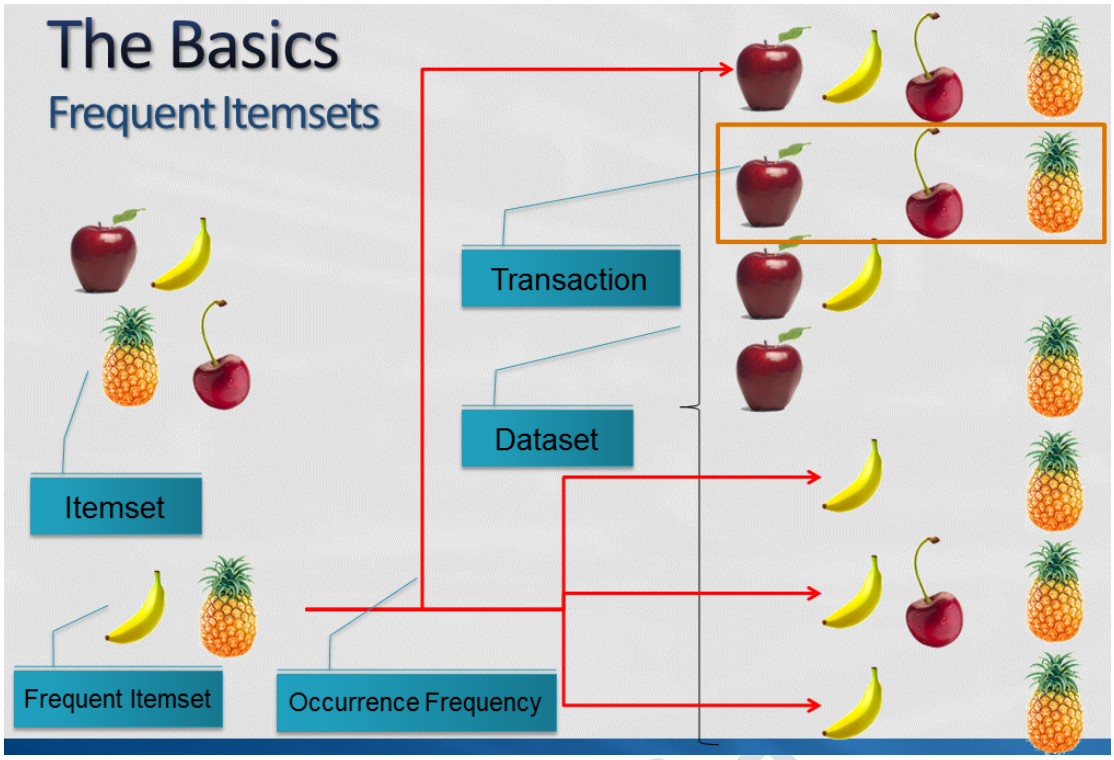
---

كل صف من الصورة بيمثل transaction

كل الانواع الموجوده هي itemsets

كل الصفوف مع بعض بيمثل Dataset

عدد مرات ظهور الحاجات اللي بيدورعليها مع بعض بتسمي Occurrence Frequency




عدد تكرار العنصرين مع بعض اكتر من threshold معين هو frequent itemset يعني الموز والاناناس اتكرروا كام مرة لو انا عامل threshold=2 يبقي عشان اقول عليهم frequent itemset لازم يكونوا اتكرروا مرتين او اكتر

---

$$support(A \Rightarrow B) = P(A \cup B) = \frac{n(A \cup B)}{N}$$

Support هو عدد المرات اللي اتكرر فيها item معينه علي عدد itemsets كلها يعني هنا عدد مرات تكرار الموز والاناناس علي عدد transaction كلها هايبقي 4/7=57%

$$support(banana \Rightarrow pineapple) = P(banana \cup pineapple) = \frac{4}{7} = 57\%$$

---

$$confidence\ (A \Rightarrow B) = P(B|A) = \frac{n(A \cup B)}{n(A)}$$

Confidence هو كام واحد من اللي اشتري item الاول اشتري item التاني يعني كام واحد اشتري موز اشتري معاه اناناس هنلاقي عدد النا اللي اشترت الاتنين مع بعض ٤ وعدد الناس اللي اشترت الموز ٥

$$confidence\ \left(banana \Rightarrow pineapple\right) = P(pineapple|banana) = \frac{4}{5} = 80\%$$

---

*min_support count* اقل عدد اتكرر عشان اقول عليه itemset

*min_support* and *min_confidence* انا اللي بحددهم علي حسب احتياجي

- If a rule satisfies *min_support* and *min_confidence* thresholds, it is said to be **strong**

لو rule اللي كنت حددتها قبل كده اكبر من *min_support* and *min_confidence* thresholds او بتساويهم بيتقال عليها strong

- *problem of mining association rules reduced to mining frequent itemsets*

- Association rules mining becomes a two-step process:

  1. Find all **frequent itemsets** with frequently ≥ a predetermined *min_support count* ودي تعتبر اكتر خطوه مكلفه

  2. Generate **strong association rules** from the frequent itemsets that satisfy *min_support* and *min_confidence*

- If *min_support* count is set too low → huge # of frequent itemsets

لو رقم *min_support* count قليل هايبقي عندي عدد كبير جدا من frequent itemsets

---

## Mining Frequent Itemsets

### 1) Apriori Algorithm

بعتمد علي حاجه بتكون كوجوده مسبقا قبل ماطبق الالجوريزم

لو عندي item مش frequent لو زودت عليه item تاني مش هايغير من حالته بمعني

A C

A C D

B

لو threshold=2 يبقي B مش frequent فلو زودت اي حرف تاني جنب B وليكن C مش هايخليها frequent

١. هابص علي كل item لوحد اتكرر كام مرة
٢. هعمل generate لحاجه اسمها candidate itemset

To improve efficiency, use the **Apriori property:**

"*All nonempty subsets of a frequent itemset must also be frequent*" – if a set cannot pass a test, all of its supersets will fail the same test as well
**if *P*(*I*) < *min_support* then *P*(*I* ∪ *A*) < *min_support***

---

**Level 1**

Transactional data example
*N*=9, *min_supp count*=2

| TID | List of items |
|---|---|
| T100 | I1, I2, I5 |
| T200 | I2, I4 |
| T300 | I2, I3 |
| T400 | I1, I2, I4 |
| T500 | I1, I3 |
| T600 | I2, I3 |
| T700 | I1, I3 |
| T800 | I1, I2, I3, I5 |
| T900 | I1, I2, I3 |

Scan dataset for count of each candidate

$C_1$

| Itemset | Support count |
|---|---|
| {I1} | 6 |
| {I2} | 7 |
| {I3} | 6 |
| {I4} | 2 |
| {I5} | 2 |

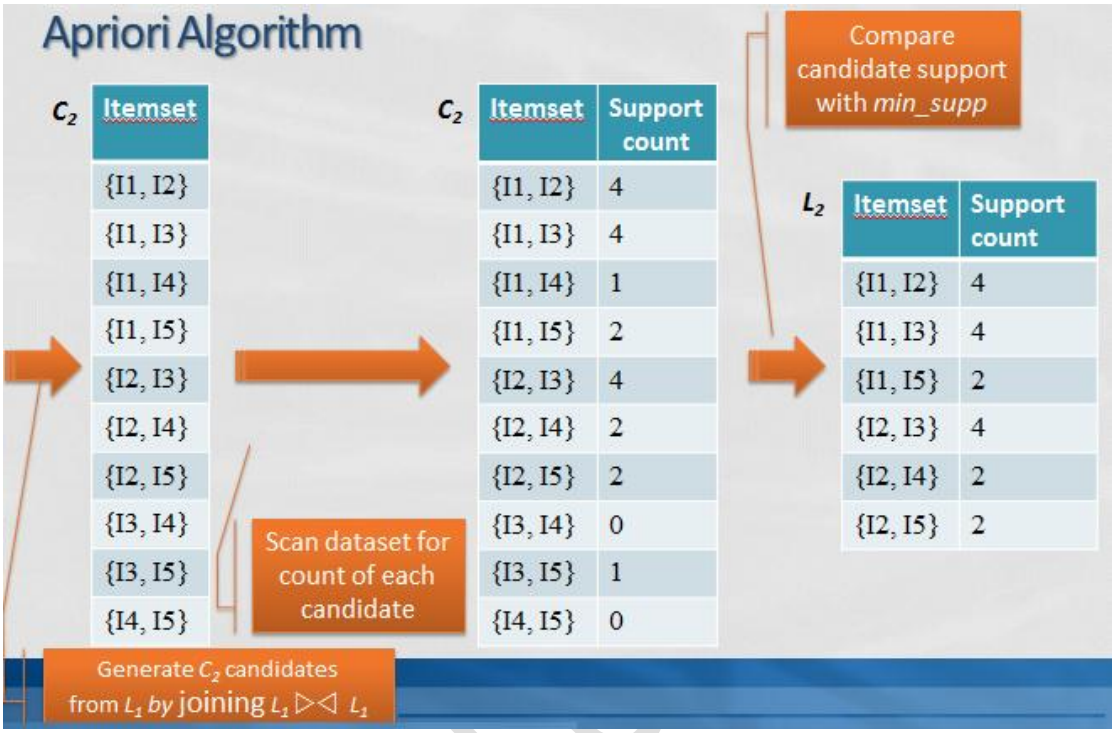
Compare candidate support with *min_supp*

$L_1$

| Itemset | Support count |
|---|---|
| {I1} | 6 |
| {I2} | 7 |
| {I3} | 6 |
| {I4} | 2 |
| {I5} | 2 |

١. اول جدول هو كل Transactions اللي عندي
٢. تاني جدول بشوف كل item اتكرر كام مرة يعني I1 اتكررت ٦ مرات في , T100 , T400 , T500 T700 , T800 ,T900
٣. تالت جدول اي Transactions اقل من threshold=2 اشيله هلاقي هنا كلهم اكبر من او يساوي ٢

**Level 2**



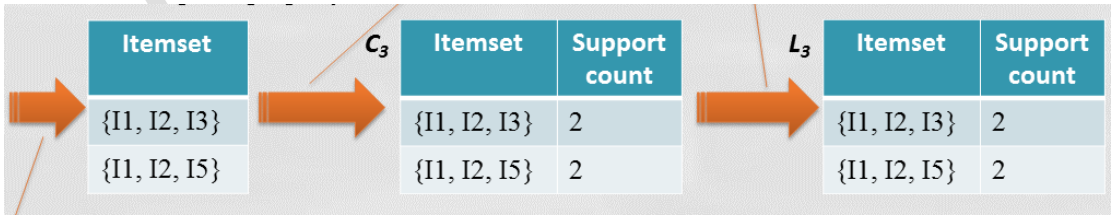

ببدا اخد كل اتنين مع بعض واشوفهم اتكرروا مع بعض:-

١. اول جدول بعمل تباديل وتوافيق لكل item مع الباقي
٢. تاني جدول بشوف كل اتنين item اتكرروا قد ايه في الجدول الاصلي اللي فيه transactions
٣. بعد كده بشيل اي itemset اتكرروا اقل من ٢

**Level 3**

باخد كل صفين من الجدول اللي في ليفل ٢ واعملهم join بس لازم يكونوا مشتركين في اول item يعني اخد I1,I2 مع I1,I3 فهما مشتركين في I1 وهو اول item فيهم بيقي اعملهم join وارتبهم

في العموم لما بختار كل join بيبقي لازم يكونوا مشتركين في كل items ماعدا الاخير




١) بعد مابختار كل صفين مشتركين في اول item هلاقيهم
{{I1, I2, I3}, {I1, I2, I5}, {I1, I3, I5}, {I2, I3, I4}, {I2, I3, I5}, {I2, I4, I5}}

بس مش هاخد منهم الا {I1, I2, I3}, {I1, I2, I5} بس لان جميع عناصرهم frequent يعني I1,I2 و I1,I3 و i2,i3 والتاني {I1, I2, I5} هلاقي I1,I2 و I1,I5 و i2,i5 هما اللي frequent (اللي support count)اكبر من 2

{I1, I3, I5}, {I2, I3, I4}, {I2, I3, I5}, {I2, I4, I5} هاحذفهم بسبب Not all subsets are frequent اول واحد
مثلا {I1, I3, I5} I1,13 و I1,I5 => frequent بس I3,I5 مش frequent وهكذا في الباقي

---

**Level 4**


مش هلاقي حاجه frequent فاهقف علي اخر حاجه كنت فيها frequent في level 3

| Itemset |
|---|
| {I1, I2, I3} |
| {I1, I2, I5} |

## Generating Association Rules from Frequent Itemsets

Association rules can be generated using the *confidence equation*, as follows:

عشان احسب association rule من frequent item بحسبها بالقانون ده

$$\boldsymbol{confidence}\ (\boldsymbol{A} \Rightarrow \boldsymbol{B}) = \boldsymbol{P}(\boldsymbol{B}|\boldsymbol{A}) = \frac{\boldsymbol{support\_count}(\boldsymbol{A} \cup \boldsymbol{B})}{\boldsymbol{support\_count}(\boldsymbol{A})}$$

1. For each frequent itemset $l$, generate all nonempty subsets of $l$
2. For every nonempty subset $s$ of $l$, output the rule:

$$\boldsymbol{s} \Rightarrow (\boldsymbol{l} - \boldsymbol{s}) \qquad if\ \frac{support\_count(l)}{support\_count(s)} \geq \min\_confidence$$

هاخدكل صف في اخر جدول الجدول مثلا الصف التاني وهي {I1,I2,I5} هابدا اقسمها ل subset واطلع منها association rule هاخد اول عنصرين{I1,I2}بيشاوروا علي العنصر التالت I5 وهكذا زي الرسمه

| Nonempty subsets | Association Rules |
|---|---|
| {I1, I2} | {I1, I2} ⇒ I5 |
| {I1, I5} | {I1, I5} ⇒ I2 |
| {I2, I5} | {I2, I5} ⇒ I1 |
| {I1} | {I1} ⇒ {I2, I5} |
| {I2} | {I2} ⇒ {I1, I5} |
| {I5} | {I5} ⇒ {I1, I2} |

هاحسب confidence بالقانون اللي فوق اللي هو $\text{confidence}\ (A \Rightarrow B) = P(B|A) = \frac{\text{support\_count}(A \cup B)}{\text{support\_count}(A)}$

في البسط عدد ظهور كل العناصر مع بعض وفي المقام عدد ظهور nonempty subsets يعني علي اول واحده عدد ظهور{I1,I2,I5} علي عدد ظهور {I1,I2} في الجدول الاصلي اللي في الشمال تحت عدد مرات ظهور العناصر كلها =٢ اللي هما T100 و T800 وعدد مرات ظهور I1,I2 =4 اللي هما T100 , T400 , T800 , T900

$$\text{confidence}\ \left(A \Rightarrow B\right) = P(B|A) = \frac{2}{4} = 50\%$$

| Itemset |
|---|
| {I1, I2, I3} |
| {I1, I2, I5} |

وهكذ لباقي العناصر في الجدول وبعد ما احسب confidence اشوف *min_confidence* واحذف اي confidence اقل منه

*min_confidence* = 70%

| TID | List of items |
|---|---|
| T100 | I1, I2, I5 |
| T200 | I2, I4 |
| T300 | I2, I3 |
| T400 | I1, I2, I4 |
| T500 | I1, I3 |
| T600 | I2, I3 |
| T700 | I1, I3 |
| T800 | I1, I2, I3, I5 |
| T900 | I1, I2, I3 |

| Nonempty subsets | Association Rules | Confidence |
|---|---|---|
| {I1, I2} | {I1, I2}⇒ I5 | 2/4 = 50% |
| {I1, I5} | {I1, I5} ⇒ I2 | 2/2 = 100% |
| {I2, I5} | {I2, I5} ⇒ I1 | 2/2 = 100% |
| {I1} | {I1} ⇒ {I2, I5} | 2/6 = 33% |
| {I2} | {I2} ⇒ {I1, I5} | 2/7 = 29% |
| {I5} | {I5} ⇒ {I1, I2} | 2/2 = 100% |

ودول اللي هاخدهم من الجدول هنا لان confidence اكبر من 70%

| | |
|---|---|
| {I1, I5} ⇒ I2 | 2/2 = 100% |
| {I2, I5} ⇒ I1 | 2/2 = 100% |
| {I5} ⇒ {I1, I2} | 2/2 = 100% |

الالجوريزم ده شغال كويس بس لو عندي item كتير هاتبقي مشكله فهاستخدم الجوريزم FP-Growth

---

## 2) FP-Growth

- To avoid *costly candidate generation*

وهاستخدمه عشان مش كل شوية اعمل candidate

- Divide-and-conquer strategy:

1. *Compress* database representing frequent items into a **frequent pattern tree (FP-tree)** – *2 passes over dataset*
2. *Divide* compressed database (FP-tree) into *conditional databases,* then mine each for frequent itemsets – *traverse through the FP-tree*

باخلي transaction اللي في الداتابيز علي شكل tree عشان اقلل ال runtime

هاعمل scan للداتابيز عشان اشوف كل item اتكرر قد ايه وبعد كده هارتبهم علي تنازليا بحسب عدد مرات ظهور كل item

Transactional data example
*N*=9, *min_supp count*=2

| TID | List of items |
|---|---|
| T100 | I1, I2, I5 |
| T200 | I2, I4 |
| T300 | I2, I3 |
| T400 | I1, I2, I4 |
| T500 | I1, I3 |
| T600 | I2, I3 |
| T700 | I1, I3 |
| T800 | I1, I2, I3, I5 |
| T900 | I1, I2, I3 |

Scan dataset for count of each candidate

$C_1$

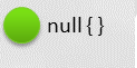
| Itemset | Support count |
|---|---|
| {I1} | 6 |
| {I2} | 7 |
| {I3} | 6 |
| {I4} | 2 |
| {I5} | 2 |

Compare candidate support with *min_supp*

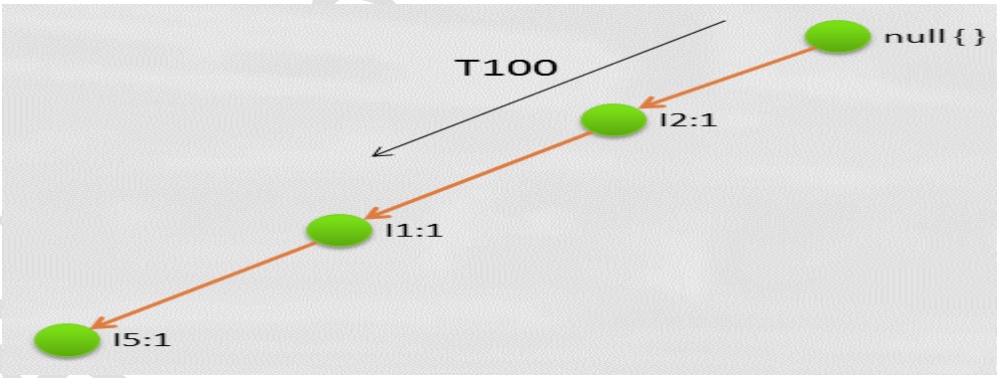
$L_1$ - *Reordered*

| Itemset | Support count |
|---|---|
| {I2} | 7 |
| {I1} | 6 |
| {I3} | 6 |
| {I4} | 2 |
| {I5} | 2 |

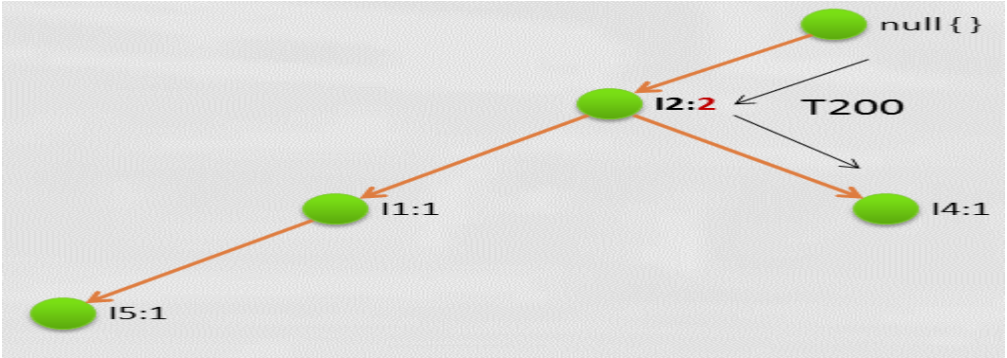
هاشوف اول item اللي هو I1 اتكرر كام مرة هلاقيه اتكرر 6 مرات T100, T400, T500, T700, T800, T900 وهكذا في كل العناصر وارتبهم من الكبير للصغير بس لو فيه حاجه اتكررت اقل من min_supp count اشيلها

اول حاجه هاعمل root ب null 

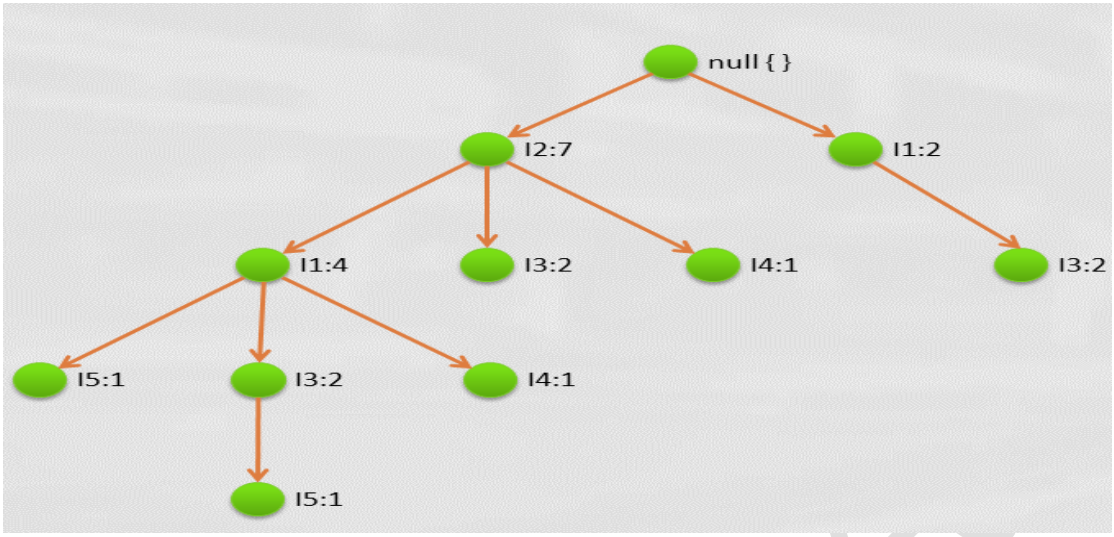
وبعد كده اخر كل صف في الجدول الاصلي ارسمه اول صف هو T100 اللي عناصره I1,I2,I5 ارتبهم زي ماقلنا هايبقي I2 ثم I1 ثم I5 وابدا ارسمهم اوصل I2 بtree واوصل I1 ب I2 وI5 ب I1 واكتب جنب كل item هو اتكرر قد ايه وهايبقي كده

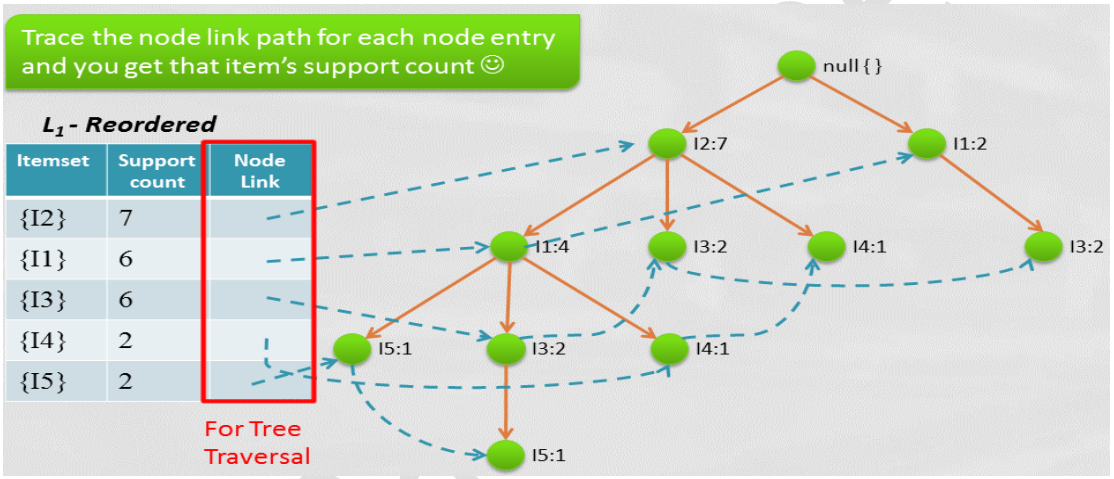



تاني صف T200 اللي هو I2,I4 هازودهم علي الرسمه ارتبهم الاول هايبقي I2 ثم I4 اشوفهم هل هما موجودين علي الرسمه او اي item فيهم هلاقي I2 هانزل منها سهم ل I4 وازود عدد تكرار I2 الي I2:2 وهاتبقي كده



وهكذا لحد مالخلص كل الجدول وفي الاخر هاتكون tree كده

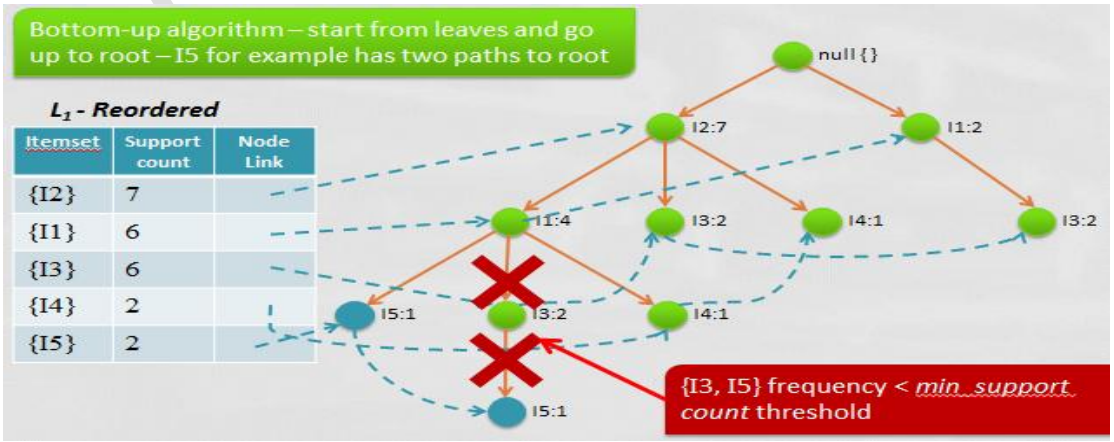



كل item في tree هاشاور عليه في node link يعني I2 موجوده مرة واحده ممكن تكون اتكررت اكتر من مرة بس مرسومه مرة واحد اشاور من node link علي النقطه دي I1 اترسمت مرتين يبقي اطلع سهم من node link بتاع I1 بيمر بكل نقط I1 وهكذا في كلهم

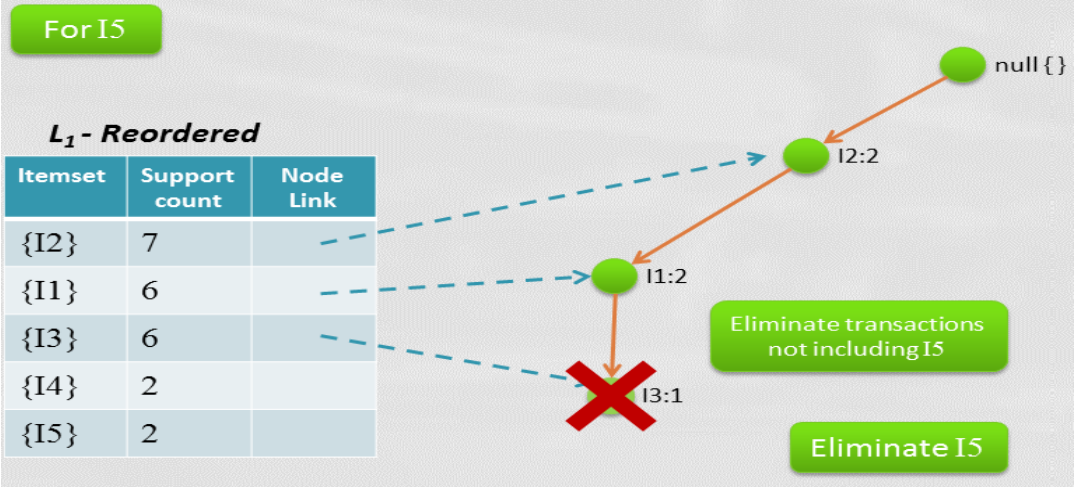
الالجوريزم ده Bottom-up algorithm يعني هابدا باخر نقطه تحت خالص اللي هي I5 لحد لما اوصل للroot

I5 هاشوف frequency هلاقيها اقل من threshold *min_support count* فهاشيله



هاعمل Conditional FP-tree Construction لكل node لوحدها

هاشيل كل transaction اللي مفيهاش node دي ولو تكن اول حاجه i5 هايتبقي T800 , T100 هاشيل منهم I5 وارسم اللي باقي فهلاقي بعد مامسحت I5 T100(I1,I2) و T800(I1,I2,I3) هارسمهم مع بعض زي ماعملت فوق وامسح اي node اتكررت اقل من min support




I2 اكتر item اتكرر مرتين مرة مع T100 و مرة مع T800 وكذلك I1 بس I2 في الترتيب الاول كانت رقم واحد فهنا هاتبقي الاول وبعد كده I1 وبعدهم I3 اللي اتكررت مرة واحده بس في T800 فهانشيلها لانها اقل من min support

1) Condition pattern base

وده الاحتمالات الممكنه قبل مامسح I3

2) Condition FP-tree

وده node اللي موجوده بعد الحذف وعدد تكرار كل واحده

3) Frequent patterns generated

ودي المسارات الممكنه عشان اروح ل I5 بعد ماشيلت I3

وامسك كل item واعمل عليه نفس العمليه دي وهايطلع الجدول ده

| Item | Conditional Pattern Base | Conditional FP-tree | Frequent Patterns Generated |
|---|---|---|---|
| I5 | {{I2, I1: 1}, {I2, I1, I3: 1}} | ⟨I2: 2, I1: 2⟩ | {I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2} |
| I4 | {{I2, I1: 1}, {I2: 1}} | ⟨I2: 2⟩ | {I2, I4: 2} |
| I3 | {{I2, I1: 2}, {I2: 1}, {I1: 2}} | ⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩ | {I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2} |
| I1 | {{I2: 4}} | ⟨I2: 4⟩ | {I2, I1: 4} |

Paths to which item is suffix

Prefix paths to item after eliminating infrequent items

## Pattern Evaluation Methods

- Not all association rules are interesting

مش شرط كل association rule اللي طلعتها تبقي مفيده ومهمه ممكن مايكونش ليها لازمه

  - buys(X, “computer games”) ⇒buys(X, “videos”) [40%, 66%]
  - P(“videos”) is already 75% > 66%
  - The two items are negatively associated → buying one *decreases* the likelihood of buying the other

- We need to measure “*real strength*” of rule
- **Correlation analysis**

Correlation ← الاتنين item بيزيدوا مع بعض او بيقلوا مع بعض اوا ملهمش علاقه ببعض

$$A \Rightarrow B\ [support,\ confidence,\ correlation]$$

عشان اعمل evalution لل asscociation rule اللي طلعتها بازود parameter كمان وهو correlation بلاضافه الي support و confidence

1. $lift = \frac{P(A \cup B)}{P(A)P(B)}$

   - *A* and *B* are *independent* if $P(A \cup B)=P(A)P(B)$

   لو $P(A \cup B)=P(A)P(B)$ يبقي قسمتهم بتساوي ١ ويبقي A,B مش معتمدين علي بعض

   - Otherwise, **dependent and correlated occurrence**

   لو مش بيساوي واحد يبقي معتمدين علي بعض وفيه correlation مابينهم

   - If $lift$ < 1, *A* is negatively correlated with *B*

   لو الناتج اصغر من ١ يبقي *A* is negatively correlated with *B*

   - If $lift$ > 1, *A* is positively correlated with *B* → *A*’s occurrence “**lifts**” the occurrence of *B*

لو اكبر من ١ يبقي , *A* is positively correlated with *B* → *A*’s occurrence “**lifts**” the occurrence of *B*

2. $\chi^2$ → already discussed in a previous lecture

تاني حاجه chi square ودي خدناها في قبل كده